可编程逻辑系统的 VHDL 设计技术

【美】Kevin Skahill 编著
(Cypress Semiconductor)
朱明程 孙 普 译
朱明程 审校

VHDL *for* PROGRAMMABLE LOGIC

东南大学出版社

可编程逻辑系统的 VHDL 设计技术

VHDL for Programmable Logic

【美】Kevin Skahill 编著
(Cypress Semiconductor)
朱明程 孙 普 译
朱明程 审校

东南大学出版社

内 容 提 要

《VHDL for Programmable Logic》一书是美国 Cypress 半导体高级技术专家编写，广泛用于美国的大学内有关电子类 VHDL 专业教学和社会上工程师 VHDL 技术培训。该书基于可编程逻辑系统的 VHDL 设计方法，系统地介绍了 CPLD、FPGA 原理、VHDL 技术概念、系统设计方法，是全球首册以 CPLD、FPGA 为目标载体，介绍 VHDL 设计技术的正式出版物，经美国多所重点院校有关专业使用，反映较好。FPGA 技术是 20 世纪 90 年代电子应用、通信技术等产业的技术应用热点，VHDL 硬件描述语言则更是专业领域普遍看重和推荐的电子系统硬件设计语言标准。本书第 1～2 章主要介绍电子设计技术概况和现场可编程逻辑器件的原理、特性及分类；第 3～9 章介绍 VHDL 设计的基本概念、语句、语法及设计风格，CPLD、FPGA 应用设计实例及技巧；第 10 章介绍 VHDL 测试平台。本书的出版，对于我国 VHDL 设计技术的普及，将是一个很好的推动。

本书可作为电子、通讯、计算机及自动控制类的大学本科和研究生专业课程教材，也适用于 VHDL 技术的工程师继续教育。

我社接受美国 Addison－Wesley 出版公司和 Cypress 半导体公司的授权，出版《VHDL For Programmable Logic》一书中文简体版。

图书在版编目(CIP)数据

可编程逻辑系统的 VHDL 设计技术／(美)斯凯希尔(Skahill,K.)编著；朱明程，孙普译. －南京：东南大学出版社，1998.9(1999.11 重印)

书名原文：VHDL for Programmable Logic

ISBN 7－81050－379－0

Ⅰ.可… Ⅱ.①斯… ②朱… ③孙… Ⅲ.可编程序控制器－程序设计 Ⅳ.TP332.3

中国版本图书馆 CIP 数据核字(1999)第 42574 号

东南大学出版社出版发行

(南京四牌楼 2 号 邮编 210096)

出版人：宋增民

江苏省新华书店经销 江苏省地质测绘院印刷厂印刷

开本 787mm×1092mm 1/16 印张 24.75 字数 630 千

1998 年 11 月第 1 版 2000 年 12 月第 3 次印刷

印数：7001－9000 定价：32.00 元

译 者 序

20世纪90年代，集成电路产业销售额增长最快的产品是现场可编程逻辑集成电路。随着深亚微米半导体工艺技术的发展，现场可编程逻辑集成电路的单片集成度日益增大，速度不断提高，价格大大下降，推动着现场可编程集成电路的应用迅速普及。

20世纪90年代，引起数字系统设计方式发生突破性变革的技术是VHDL设计技术。VHDL[Very High Speed Integrated Circuit (VHSIC) Hardware Description Language]作为IEEE－1076标准所规范的硬件描述语言，随着各种EDA工具和集成电路厂商的普遍认同和推广，目前正在全球范围内先进工业国家的电子系统设计领域获得广泛应用。

深圳大学EDA技术中心作为深圳市科技局资助设立的、构架于大学教学科研环境中、为社会电子产业界服务的开放式电子设计基地，其要旨之一就是跟踪国际电子设计领域最新技术，率先消化吸收、继而转化推广，为促进中国电子应用产业界的技术进步、产品设计的更新换代作出贡献。3年来，我们除了在FPGA技术应用、MCU用户电路设计等方面常年为社会提供技术培训和开放式项目设计服务外，一直期望尽快将VHDL设计技术引入国内，尽快为我们大学的电子信息工程、通讯工程、计算机科学与技术、自动控制、微电子学等专业的本科生、研究生以及相关企业中的电子设计工程师创造一种方便易学、理论与实验相融合的教学与实践环境，使VHDL设计技术尽快在我国推广和普及。

为此，我们接受美国Cypress半导体公司的委托和授权，将其拥有版权的由Mr.Kevin Skahill编著的《VHDL For Programmable Logic》一书译成中文版本，在国内出版发行。该书的原英文版本在美国出版后，受到美国众多读者和工程师的欢迎，已成为美国多家大学的有关VHDL课程的教材。该书的最大特点是，以广泛应用的Cypress CPLD和FPGA为目标载体，以Warp2 VHDL设计工具(软件)为手段，介绍VHDL语言及电路设计。因其以典型数字系统设计过程为线索，使读者在循序渐进的练习和设计实验中，理解和学会用VHDL来进行数字系统的设计，并可通过现场的CPLD或FPGA硬件实现来验证其系统的功能和性能（有关设计软件的支持，可向译者咨询，E－mail:zhumc@szu.edu.cn）。该书附有大量的练习和设计实例。由于本书完全以介绍VHDL设计技术为主题，故所有支持VHDL的设计软件均可辅助读者的学习和实践。

本书由深圳大学信息工程学院朱明程副教授及其助手孙普（目前在英国利物浦大学研读深造）翻译，其中第1～5章及部分附录由朱明程翻译，第6～10章及部分附录由孙普翻译。全书由朱明程审校、修改、定稿。译者在翻译过程中，对原书中的少量错误作了修正。同时，承东南大学无线电系黄正瑾和宋继亮老师协助进行了印刷样稿的审校，香港庆成企业有限公司中国部应用工程师李远辉、李昆华先生协助进行了部分习题的设计验证，在此一并致以谢忱。

在本书的翻译过程中，我们力求译文准确、风格统一、概念规范、通俗易懂，但由于时间仓促，同时限于译者水平，不足和错误之处在所难免，恳请读者不吝指正。

在本书付诸出版之际，译者感谢美国Cypress半导体公司的信任和支持，感谢香港Tekcomp公司的推荐和协调，感谢深圳大学名誉教授、中科院半导体研究所王守觉院士，北京理工大学ASIC技术研究所所长刘明业教授，以及我们学院的刘家祥教授等老师的鼓励和指导。

译 者

1998年8月于深圳大学

序　言

《可编程逻辑系统的 VHDL 设计技术》这一本书适合作为有关 VHDL 语言、数字逻辑设计、ASIC 设计课程的教材，它可用于大学高年级学生、研究生教学及电子设计工程师技术培训，以提供和更新其采用 VHDL 和可编程逻辑器件的电子设计方法学方面的知识和技术内容。

本书的目标是希望提供给读者：

(1)掌握正确进行逻辑电路综合的 VHDL 格式文件的写作方法；

(2)理解 VHDL 和可编程逻辑设计的过程——从设计描述、综合、布局布线，到设计和产生检验用的测试结构；

(3)设计因素综合考虑的知识；

(4)能够升级和用来解决各种设计问题的工具技术。

为达到如上目的，本书在详介 VHDL 设计方法的过程中，侧重于：

(1)介绍抽象理论，同时注重强调实例；

(2)有关设计讨论的一些主题：状态机设计、性能与面积的权衡、参数化元件、阶层结构与库、布线通路与资源分配；

(3)采用多种实例描述通过综合方法化简到逻辑门的过程；

(4)有关 CPLD 和 FPGA 的数字电路综合、装配、布局布线的示范；

(5)围绕设计主题的专用练习题，将对读者使用 Warp 工具起到促进作用；

(6)附在 CDROM 中超过 80 个设计实例文件，可供解决常见的实际问题作参考，并由此提高手工进行综合、装配、布局布线的经验。

本书的写作要旨在于通过讨论涉及读者常见的综合和设计问题来学习一种新的设计语言和设计方法，通过一系列设计问题、实例和技术上的描述，教会读者采用可编程逻辑器件来实现 VHDL 设计模型的综合结果。这种方式不同于介绍语法和语言结构的做法，那种趋于强调构造但不一定能够有效综合的模拟模型的抽象和理论近似，而这种设计往往不是在物理上无意义，就是其语句结构不适于综合。

为了缩短理论到实践的距离，书中采用许多练习来促进读者对综合软件的运用。它可以帮助读者了解抽象的 VHDL 描述如何被化简成数字逻辑。实际上可望使读者学会编写能够被综合成有效的数字逻辑电路的 VHDL 格式文件。

如何理解和处理设计目标也在书中给予强调，因为读者在实际设计中，必须对诸如产品上市时间、成本、设计特性、工作性能及生产能力等需求作出抉择和权衡。

本书的构架着重于综合和设计方面，为了保持这个特点，将有关测试的章节放在最后。凡是希望尽早学会测试结构写作的读者，可在学完第 4 章后提前学习第 10 章有关内容。这样将使读者既能用 VHDL 去编写设计，又能编写测试结构，以完成书中的练习。

说明性的资料

软件(Software)：其中包括 Cypress 半导体的 Warp 2 CPLD 和 FPGA 的综合、装配、布局布线的软件，也包括一个交互式波形模拟器，可用于进行基于 JEDEC 编程文件的功能模拟。

Warp 软件的使用说明集中于用户说明中，但它并不等同于软件的使用。对于具有其他 VHDL 工具使用经验的读者，阅读此书较为容易。但是，为了适应读者特定的软件环境。书中的例题和章节末尾的习题则往往需要进行适当修改。

解题手册(Solution Manual)：对于每一章末的的解题答案，使用者可通过 Addison - Wesley 的 ftp.aw.com/cseng/authors 或者 Cypress 的 http://www.aw.com 查寻。

建议

尽管在原稿的不同阶段，审阅者都进行了审阅和指正，但书中还难免有错误存在。如果您在阅读时，发现什么错误或有什么改进的建议，请用 E - mail 见告。出版社的 E - mail 地址：aw.eng@aw.com。作者的 WWW：http://www.aw.com。同时欢迎提供新的实例和练习(请附答案)。

致谢

对于在这个项目全过程中给予我帮助的人们谨此表示诚挚的谢意。我认为没有他们的帮助是无法完成这本书的写作的。所以，首先我得感谢 Barry. Fitzgerald，他的鼓励使此书顺利落稿。一经落稿，基本上就依靠 David Johnson 和 Terri Fusco 来处理手稿和征求其他的支持。特别要感谢的是 David Johnson，是他帮助制作了书中几乎所有的图稿，并且为我完成这部书提供了许多指导和帮助。

作　者

于美国加尼福尼亚州

目　录

1 概 述

20世纪90年代上半叶，由于个人电脑、无绳电话和高速数据传输设备的发展需求，电子工业经历了巨大的飞跃。为了赢得商业上的竞争，生产商的产品迫切需要追求高功能、优品质、低成本、微功耗和微小封装尺寸。为此，生产商必须采用少量的IC器件和面积尽可能小的PCB板来研制高集成化的复杂系统。亚微米半导体工艺、PCB表面安装技术的发展支持了产品的集成化程度的进步。但是，在给定EDA工具的条件下，随着产品上市时间周期的加速，设计复杂程度的提高，影响生产商开发的瓶颈问题就是其设计能力，这个状况实际上孕育着对现代设计方法和现代测试方法的普遍需求。如此，高密度逻辑器件和VHDL(超高速集成电路硬件描述语言)成为解决这些问题的关键所在。

高密度现场可编程逻辑器件，包括CPLD和FPGA，能够将大量逻辑功能集成于一个单片IC之中。虽然半定制和全定制的专用集成电路(ASIC)也能够实现将大量数字逻辑功能集成于单片之中，但CPLD和FPGA具有更多的灵活性：既适用于短研制周期、小批量产品开发，也可用于大批量产品的样品研制。同时，因其项目开发所需前期工程开发费用低的特点，更有着诱人的应用前景。

VHDL非常适用于可编程逻辑器件的应用设计，并正在得以普及。在500～10×10^4门的大容量CPLD和FPGA的应用设计中，工程师若采用以往的布尔方程或门级描述方式，难以快速和有效地完成设计。而VHDL却能够提供高级语言结构使工程师很方便地描述大型电路，促进产品的快速上市。它能够支持设计单元库的创建，以存储在附属子设计中重复使用的元件。因为VHDL是一种标准语言(IEEE1076标准规定)，在综合和模拟工具之间，VHDL代码具有可移植能力，即设计可用不同的器件来实现。同样，采用VHDL，实现一个设计从可编程器件向ASIC的转换也是便利的。

就小规模的设计灵活性和集成度而言，低于500门的简单PLD经常被成功地采用。其传统的设计技术，诸如卡诺图，常用来生成在PLD中实现设计功能的设计方程。使用具有一定语法规则的简单语言来形成组合和寄存形式方程，设计者首先以数据文件的格式着手设计方程，然后由软件来进行方程的综合，最后形成一个用于PLD器件编程的数据文件。

对于采用CPLD、FPGA或者ASIC的大型系统，上述的传统设计方式是行不通的，传统技术的生成逻辑方程的方式既费时、又易出错，且在方程式中查寻错误也很困难。而图形输入方式则有许多优点，例如：可以提供设计的图形观察，具有支持图形阶层结构的软件工具，使设计构成模块化形式。但是，对于大型复杂的设计，纯图形输入方式也是有其弊端的。

●控制逻辑往往仍必须用传统设计技术来产生。

●原图的保持比较困难，在设计实现过程中，经常需要对设计进行修改，同时，在实现过程中，设计构图的形式也会改变。

●图的方式经常需要附一个文本来描述其设计构思和功能，用英语或其它语言形式以能

够对用户提供设计解释。

●图形输入的环境是专用的,这使得设计者在某个图形输入环境中工作的某个项目设计,其设计的图或子图,无法在另外一个图形输入环境中进行另一个项目设计时重新调用。

●PLD 图形输入工具支持的模拟环境不一定适合系统设计环境,故进行设计验证往往是困难的。

一个较优选的设计方式应该能够提高设计者的工作效率。较详细地来概括,它应该能促进设计输入、设计理解、设计维护的便利和快捷。它即便不依赖于解释,也应能较方便于定义。它应该是开放的、非专用的、工业界能接受的标准。它允许设计在不同的 EDA 工具环境之间移植,其模块可以封装成独立单元,重复使用。它支持阶层结构的复杂设计和从门级到系统级的设计。而且可以用于逻辑电路的描述、综合,并可以支持多层次的设计描述。

仅仅有 VHDL 和 Verilog 两种语言能够满足数字逻辑设计的这些需求。Verilog 似乎比 VHDL 更为简练,但在论述设计问题时,却未必便利。无论是文本的组合利用,还是综合,以及对器件和系统的模拟方面,VHDL 都是一个较好的选择。

VHDL 是在 70~80 年代中,由美国国防部资助的 VHSIC 项目开发的产品。在这个语言首次开发出来时,其目标仅是一个使电路文本化的一种标准,为了使人们采用文本方式描述的设计,能够被其它人所理解。同时,也被用来作为模型语言,用于采用软件来进行模拟。VHDL 于 1987 年由 IEEE 1076 标准所确认。1988 年,Milstd454 规定所有为国防部设计的 ASIC 产品必须采用 VHDL 来描述。1993 年,IEEE 1076 标准被升级、更新,新的 VHDL 标准为 IEEE 1164。1996 年,IEEE 1076.3 成为 VHDL 综合标准。

今天,VHDL 已成为一个数字电路和系统的描述、建模、综合的工业标准。在电子产业界,无论 ASIC 设计人员,还是系统级设计人员,都需要学习 VHDL 来提高他们的工作效率。由于 VHDL 所具有的通用性,它已成为可支持不同层次设计者需求的标准语言。本书通过综合和设计的实例来详细介绍 VHDL。我们将探讨一些可编程逻辑设计的 VHDL 应用,与着眼于语法、语言结构,侧重于强调不能进行综合的 VHDL 模块的抽象和理论近似的讨论相反,我们通过综合和设计来叙述 VHDL,主要基于两个理由。其一,这样处理提供了一种促进语言学习的有效途径,其强调一种积极的学习方式——即要求学员通过使用语言创建自己的设计这个过程来学习。其二,这样学习,使读者使用 VHDL 语言就如同在工作中使用一样,本书教会读者们的有关 VHDL 知识同样可应用于今后工作中的 VHDL 设计中去。同时,为了弄清设计工程师为设计综合和模拟写 VHDL 码的实质,我们的讨论还将勾画有关建模的概念。

1.1 采用 VHDL 的原因

电子工业界的每个设计工程师应该尽快学习和采用硬件描述语言,以保持能够参与竞争、设计产品的能力。使用 VHDL,你可以快速地描述和综合 5 千门、1 万门、2 万门的电路,而同类型的设计,如果采用寄存器/传输门级的图形输入或布尔方程来描述,往往需要 1 个人花几个月的工作量。此外,VHDL 提供如下所述的一些能力。

1.1.1 功能与灵活性

VHDL 具有功能强大的语言结构,可用简洁明确的代码描述来进行复杂控制逻辑的设

计。为了有效控制设计的实现，它还具有多层次的设计描述功能，支持设计库和可重复使用的元件生成，它支持阶层设计，且提供模块设计的创建。VHDL是一种设计、模拟、综合的标准硬件描述语言。

1.1.2 非依赖器件的设计

VHDL允许设计者生成一个设计而并不需要首先选择一个用来实现设计的器件。对于同一个设计描述，可以采用多种不同器件结构来实现其功能。若需对设计进行资源利用和性能方面的优化，也并不是要求设计者非常熟悉器件的结构才行。相反，你可以集中精力从事你的设计构思（当然，这并不是说设计者可以忽略诸如数据通路优化技术方面的需要，例如，在第9章中将要讨论的"流水线"技术）。VHDL也支持多种形式的设计描述，例如，图1.1所示是1个2位比较器的4种描述方式。

```
u1: xor2 port map(a(0), b(0), x(0));
u2: xor2 port map(a(1), b(1), x(1));
u3: nor2 port map(x(0), x(1), aeqb);
```

网　表

```
aeqb <= (a(0) XOR b(0)) NOR
     (a(1) XOR b(1));
```

布尔方程

```
aeqb <= '1' when a = b else '0';
```

并发语句

```
if a = b then aeqb <= '1';
   else aeqb <= '0';
end if;
```

顺序语句

图1.1　VHDL支持的多种形式的设计描述

1.1.3 可移植性

VHDL的可移植能力（Portability）是允许设计者可对需要综合的设计描述进行模拟，在综合之前对一个数千门的设计描述进行模拟可以节约设计者可观的时间。在这时发现的设计上的瑕疵就能够在设计实现之前给予纠正。因为VHDL是一个标准语言，故VHDL的设计描述可以被不同的工具所支持。从一个模拟工具移植到另一个模拟工具，从一个综合工具移植到另一个综合工具，从一个工作平台移植到另一个工作平台去执行。这意味着同一个VHDL设计描述可以在不同的设计项目中采用。在某个EDA工具中构成的技术诀窍，在其它工具中同样可以采用。图1.2所示的某个设计的原代码同样能够在其它任一综合工具上使用。并且，这个设计可以由这个综合工具所支持的任何器件来实现。

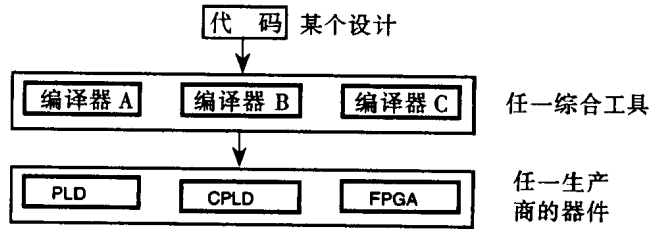

图1.2　VHDL的不依赖编译器及器件的设计可移植能力

1.1.4 性能评估能力

非依赖器件的设计（Device－Independent Design）和可移植能力允许设计者可采用不同的器件结构和不同的综合工具来评估设计。在设计者开始设计之前，无需了解将采用何种器件，是CPLD还是FPGA。设计者可以进行一个完整的设计描述，并且对其进行综合。生成选定的器件结构的逻辑功能。然后再评估结果，选用最适合你设计需求的器件。为了衡量综合的质量，同样可用不同的

综合工具所进行的综合结果，来进行分析、评估。

1.1.5 ASIC 移植

VHDL 语言的效率体现之一，就是如果你的设计是被综合到一个 CPLD 或 FPGA 的话，则可使你设计的产品以最快速度上市。当产品的产量达到相当的数量时，采用 VHDL 能很容易地帮助你实现转成 ASIC 的设计。有时，用于 PLD 的原代码可以直接用于 ASIC。并且，由于 VHDL 是一个成熟的定义型语言，用 VHDL 来设计，可以确保 ASIC 厂商交付优良品质的器件产品。

1.1.6 上市时间快、成本低

VHDL 语言和可编程逻辑很好地结合，将大大提高数字单片化设计实现速度。VHDL 语言使设计描述快捷、方便，可编程逻辑应用则将产品设计的前期风险投资降至最低，并促进设计的快速复制简便易行。同时，多种综合工具支持这种形式的设计。VHDL 和可编程逻辑的组合作为一类强有力的设计方式，将为设计者的产品上市带来创记录的速度。

1.2 不足之处

设计工程师普遍关心 VHDL 的三个问题是：(1) 电路采用高级的简明结构 VHDL 描述，它意味着放弃了对电路门级实现定义的控制；(2) 由综合工具生成的逻辑实现效果不好；(3) 工具的不同导致综合质量不一样。

对于第一个问题，很难直接给予回答。事实上，采用 VHDL 作为综合的描述语言，其目的就是希望将工程师从专门的门级电路实现的繁杂劳动中解脱出来。所以，如果你懂得编译器是怎样来综合逻辑的，你就很容易理解编译器能够优化地帮助设计实现大多数的设计构造，并且很少需要规定设计实现的规则。然而，大多数综合工具允许设计者采用综合优化指引的方式，来对实现电路和设计提供某种程度上的控制。例如，指定采用面积优化或者速度优化选择。也有一些综合工具允许设计者去专门定义设计的技术特征，实现手工从事的门级设计的工作。然而，这些形式的描述往往只能是低层的和指定器件结构来进行。

第二个问题是对于工具的逻辑综合有效性的担心，且往往不是没有根据的，VHDL 编译器并不一定总能生成满意的设计，因为优化的结果往往依赖于设计的目标。编译器采用一定的算法，由标准的设计方式来决定设计的实现，而算法从某种路径并不能发现设计中的所有问题。有时，当人们希望按自己要求来控制设计实现时，没有什么能够替代人的创造力。综合导致很差的设计实现的说法常常是由于无效的编译所引起的。诸如无效的 C 编译，会导致慢的执行时间或差的存贮器使用一样，无效的 VHDL 编译，也会导致无用的、重复的或非优化的逻辑产生 .

第三个问题往往在商业领域被提及。幸运的是，对于 CPLD 和 FPGA 的 VHDL 综合刚刚起步，其竞争会不断加强，综合技术的成熟的过程尤如 C 编译器所走过的途径一样。一段时期以前，一般 C 编译器所生成的可执行码往往无效，以致不如设计者自己去写机器码来替代。今天，仍有一些作业需采用机器码来写较为合适，但大多数代码必须用高级语言来写；C 编译器已在那些不能熟练书写机器码来编程序的工程师中广泛使用。然而，由于现在项目的程序设计的容量较大，通常还是需要他们花一定的时间于设计工作的其它方面。

1.3 采用 VHDL 设计综合的过程

通常的设计过程可划分为下述的 6 个步骤，本节先作简要介绍，在第 10 章中将着重介绍步骤(3)和(5)，而本书其余的部分将介绍步骤(2)和(4)。

(1) 设计要求的定义；
(2) 用 VHDL 进行设计描述(系统描述与代码设计)；
(3) 原代码模拟；
(4) 设计综合，设计优化和设计的布局布线；
(5) 布局、布线后的设计模块模拟；
(6) 器件编程。

1.3.1 设计要求的定义

在从事设计进行编写代码工作之前，必须先对你的设计目的和要求有一个明确的认识。例如，你要设计的功能是什么？对所需的信号建立(Setup)时间、时钟/输出时间、最大系统工作频率、关键的路径等这些需求，只要有一个明确的定义，这将有助于你的设计，然后再选择适当的设计方式和相应的器件结构，进行设计的综合。

1.3.2 采用 VHDL 进行设计描述

设计规划(Formulate the Design) 有了设计要求的定义后，你可以尝试去编写设计代码。但是，我们建议首先应决定设计方式。只有对如何描述你的设计有了一个最佳的认识，才能更有效地使你编写设计代码，然后再通过综合，进行所需要的逻辑实现。

人们通常熟悉的设计方式无非有三种：自顶向下设计、自底向上设计、平坦式设计。前两种方式包括设计阶层的生成，而后一种是将描述的电路当做单模块电路来进行的。

自顶向下的处理方式要求将你的设计划分成不同的功能元件，每个元件具有专门定义的输入和输出，并执行专门的逻辑功能。首先生成一个由各功能元件相互连接形成的顶层模块来做成一个网表，然后再设计其中的各个元件。而自底向上的设计方式恰恰相反，先定义和设计各个分立的元件，再将各元件合成，以形成总的设计。至于平坦式设计则是指所有功能元件均在同一层同一图中详细进行的。

平坦式设计对于小型设计较为适合，其功能块的基本定义内容不能分解，否则难以理解设计的功能。在本书中有许多小的设计实例，均采用平坦式设计处理方式。阶层式设计结构在包含多个复杂功能元件的大型设计中较为实用。阶层结构的层能够规定各功能元件之间的内连接关系。然而，由于过多的层将使了解设计的内连接，以及功能元件和与整个设计相关联的理解变得困难，故在设计中应尽可能地避免过多的层次。

设计代码的编写(Code the Design) 决定了设计方式以后，根据 VHDL 具体的语法和语义结构，你可以参照你已设定的功能块、数据流状态图等，来进行设计代码的编写。通常许多工程师从事设计只是简单地参照和修改一些现成的实例的方式，来实现自己设计的专门需求。但是，编写一个优化的 VHDL 代码的关键在于要依照硬件的内在要求去思考，特别是，要能像综合软件运行时的“思考”方式那样去体验如何实现你的设计。本书的一个优点在于能够帮助

你模拟综合软件的思考方式，进行优化的代码编写。

1.3.3 原代码模拟(Simulate the Source Code)

对于大型设计，采用 VHDL 模拟器进行设计的原代码模拟可以节省时间。并行工作程序(并行作业而不是串行)导致电路模拟(往往在后期作业进行的)提前至设计的早期阶段。采用原代码模拟，可以在设计的早期阶段检测到设计中的错误，从而进行修正，以便尽可能地减少对设计日程计划的影响。但对于小型设计，则往往不需要先作原代码模拟，即使做了，意义也并不大。因为对于大型设计，其综合、布局、布线往往要花费好几个小时，在综合之前进行原代码模拟，就可以大大减少设计重复和修正错误的次数和时间。当然，大型设计往往是阶层结构的序列子设计或模块的组合。模块化结构允许你在进行阶层链接之前检测和修正每个子设计，分别检测和修正各个设计单层，将可节省可观的时间。但是，通常你不必花太多的时间先去模拟原代码，因为在综合后，你往往会发现为了实现性能目标，将需要修改你的设计。在这种情况下，用户事先在原代码检测时所花的时间是毫无意义的，因为一旦设计改变，还必须重新再做检测。

1.3.4 综合、优化和装配设计

综合(Synthesis) 我们已经多次使用“综合”这个词汇，其明确的定义为——将设计描述化简到底层电路表示（诸如一个网表）。简单地说，综合是从设计描述转换到网表或方程生成的过程。这个过程也可以被解释为设计描述作为输入，而设计网表和逻辑方程作为输出。为了简单明了起见，有时也略去“输入”或“输出”的说法，直接用短语“逻辑描述综合”、“逻辑综合”、“综合逻辑”来表示。

VHDL 综合软件工具将 VHDL 描述转译成工艺专用网表或一组逻辑方程。设计者采用综合工具可直接由编写设计描述来设计逻辑电路，而不一定非要进行布尔运算或建立工艺专用的优化网表来实现。设计者只要表达清楚其设计的简要描述，定义逻辑电路的功能，综合软件工具就可以生成一组可配置于 PLD/CPLD 的逻辑方程，或者生成布局、布线于某种 FPGA 的网表。综合应该是工艺专用的。换言之，对于 VHDL 描述所进行的网表编译随着装配器件构造的不同而不同。图 1.3 所示是综合和优化的过程。在综合之前，软件工具必须阅读设计并进行分析，检测语法错误和尽可能减少语义错误。然后，综合程序将设计转换成数据结构，将设计的“行为”描述编译成“寄存器传输级(RTL)”描述。寄存器描述详细标定寄存器，信号输入，信号输出和相互之间的组合逻辑。也有一些 RTL 元件由器件结构的专用功能块来定义(例如，有些可编程逻辑器件内含 XOR 门)。在这里，组合逻辑仍由数据结构来表示。有的综合工具还将对可等同的

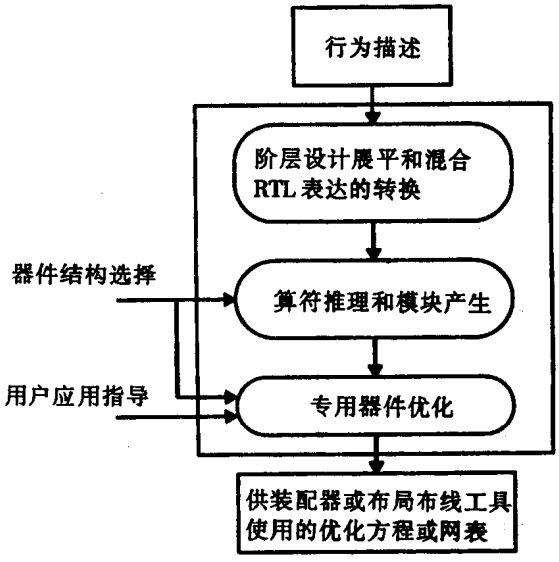

图 1.3 设计综合和优化过程

运算符和数进行数据结构的搜寻，用工艺专用的优化元件来代替这部分逻辑。这些运算符可能非常简单，等同于器件已有的 XOR 门的使用；这些运算符也可能很复杂，推断为一个 16 位的加法操作。另外还有一些不能被功能元件等同替代的逻辑，则会被转换成暂不能优化的布尔方程表达。

优化(Optimize)　优化处理依赖于三个因素：布尔表达方式，有效资源类型，以及自动的或用户定义的综合指引(有时称作约束条件)。有的表达方式有助于有效地规划逻辑资源，例如，乘积项的最小和式适于用 PAL 来实现，乘积项的标准和式更适于规划多路选择器（MUX）或 RAM。有时，标准的或最小的乘积项和式也许是用户规范的最好的表达方式。另一些用户的或自动的约束条件，可用来进行可用资源的优化表达。这些约束可用来限制表达式中字面量的出现次数(减少信号加载)，限制表达式中的字面量数(减少扇入)，或者限制表达式中项数。

对 CPLD 的优化通常包括将逻辑化简为乘积项的最小和式，为了获得最小字面量项数计数再优化。这将减少乘积项的利用，降低任何给定的表达式所需的逻辑块输入数。这些方程进一步通过器件专用优化来实现资源配置。对 FPGA 的优化通常也需要用乘积项的和式来表达逻辑。由此，方程系统可基于器件专用资源和驱动优化目标指引来实现因式分解。分解的因子可用来对实现的有效性进行评估，其准则可以用来决定是对方程系统进行不同的因式分解还是保持现有的因子。其中，准则通常是指分享共同因子的能力，即可以被暂存，以便用于和任何新生成的因子相比较。

另一种优化的方法不包含那么多布尔表达式的运算。它主要采用二元决策图方式把给定的方程规划成专用的逻辑实现。

装配(Fitting)　装配是指把通过综合和优化进程所得到的逻辑，安放到一个逻辑器件之中的过程。如果需要的话，使逻辑形式变形，以获得最优化的装配结果。“装配”一词，通常用来描述对一定的 CPLD 结构的资源进行分配的过程。布局和布线则是将综合和优化后生成的逻辑(如果需要，可使其改变形式)，规划到一个 FPGA 的逻辑结构之中(元胞阵列之中)，将各逻辑单元放置到相应优化的位置，根据信号传输的要求，在逻辑元胞之间、逻辑元胞与 I/O 口之间进行布线。为简捷起见，我们将交替使用“装配”和“布局、布线”这些词汇，以分辨上述的不同过程。

在 CPLD 中装配一个设计是一个较为复杂的过程，因为有很多路径可以将逻辑置入器件。不管如何，在以任何方式将逻辑置入器件之前，都要根据可利用的资源去进一步优化逻辑方程。例如，一些宏单元允许将一个触发器(Flip - Flop)配置为 D 型或 T 型，也允许输出极性可以选择。在这种情况下，对于 D 型和 T 型触发器原逻辑和其“反”的逻辑表示就可以生成。若实现其优化，可以从每个表达式的 4 个版本中去选择。在所有的方程变换之后，共享不足资源的表达式可以相互组合(诸如复位、预置位、时钟、输出使能、输入宏单元等)，基于用户约束，例如，脚(pin)定义的表达式也可能相互成组。往往要检测成组逻辑是否能够置入同一个逻辑模块，即要对组内集合的资源进行估价，如：所需的宏单元数目，独立的复位和预置位，输出使能，专用的乘积项，时钟(和时钟极性)，总的输入信号，总的输出信号等。如果成组逻辑的资源需求超出逻辑块的物理限制，则必须修改成组逻辑。然后再尝试将逻辑置入。如果合法放置不能成功，则需重新进行逻辑成组进程，采用和上次不同的不足资源共享方式来成组。一旦所有的逻辑元件都能找到位置，就开始尝试进行输入、输出、逻辑元胞之间的布线。如果对于给定的逻辑放置方式，布线无法成功，则布线器仍将提示你重新进行逻辑组合。如果布线成功，则装

配完成。

布局布线(Place and Route)　布局布线工具的好坏,对于 FPGA 设计的性能有着很大的影响。传导延时基本上取决于布线延时。一个"优化"的布局布线可将电路的关键部分紧密地配置在一起,以消除布线延时。布局布线工具采用一定的算法,指引用户约束和性能估价来选择最初的布局方式,然后,算法能够重复进行布局的小小改变,以逐步实现符合性能要求的优化的布局结果。接着开始布线,首先对高扇出信号或者必须长距离布线的信号采用全局布线结构来进行,然后采用局域布线结构对局域内逻辑单元和 I/O 信号通路进行布线。

图 1.4 所示的是如下相等比较器经过综合、优化被置于 CPLD 和 FPGA 的进程:

```
aeqb <= '1' when (a = b) else '0';
```

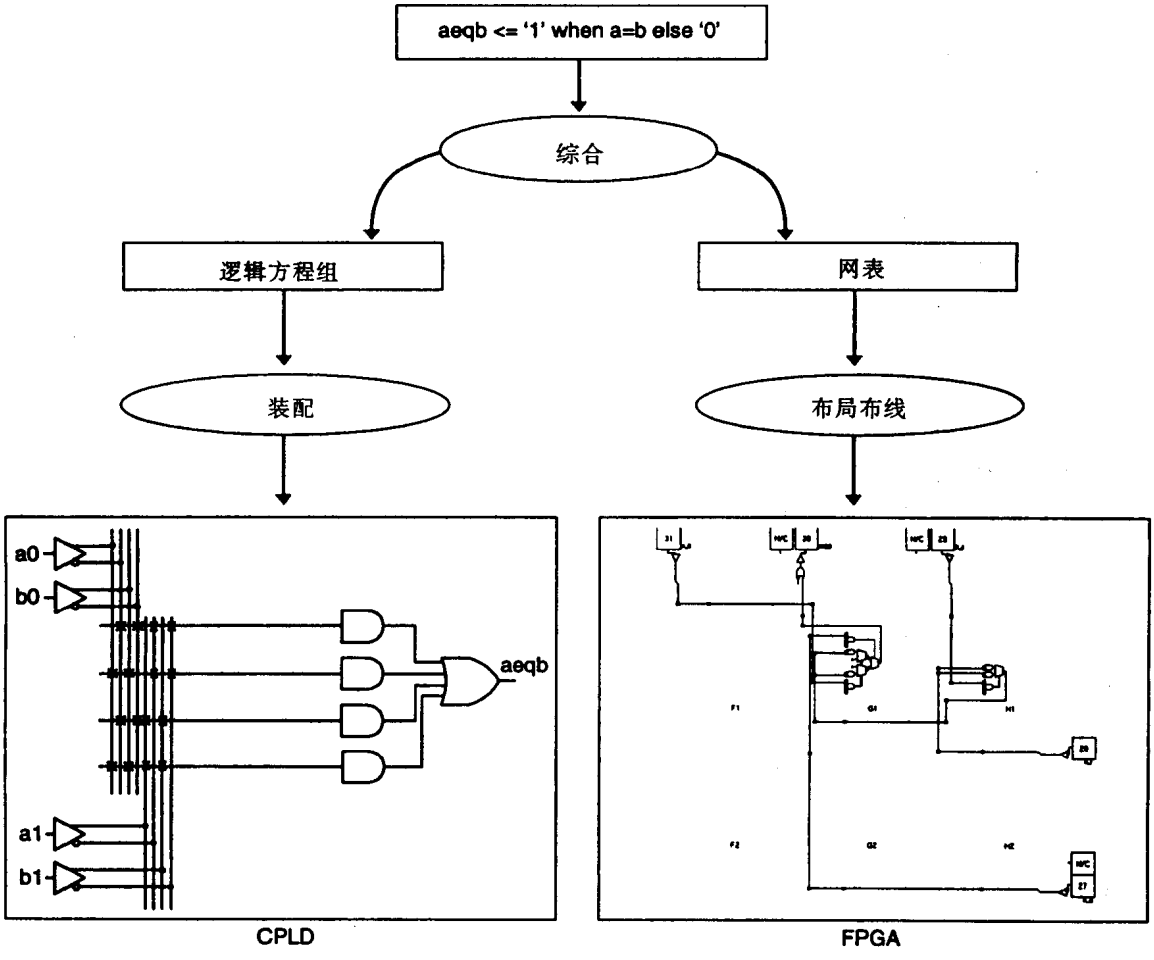

图 1.4　设计实现的过程

目前,有许多商用综合工具厂商,其中应用最普遍的有 Synopsys、Cadence、Mentor Graphics、Viewlogic、DATA I/O、Synplicity 以及一些提供自己器件专用综合的可编程逻辑器件厂商。每一个综合工具厂商采用各自不同的专有的算法进行逻辑综合。以致于采用不同工具所进行的逻辑实现的结果往往是不同的(但功能是等效的),就如同采用不同的 C 编译器编译 C 语言程序一样。尽管如此,还是有一些基本的综合规则用于不同的综合工具中,可以对综合进行有关的